Paper: jc*-**-**-****:

A Memetic Algorithm for High-Speed Railway Train Timetable Rescheduling

Shuxin Ding[*1,*2], **Tao Zhang**[*1,*2,†], **Ziyuan Liu**[*3], **Rongsheng Wang**[*1,*2,*4], **Sai Lu**[*5,*6], **Bin Xin**[*5,*6,†], **and Zhiming Yuan**[*1,*2]

[*1]Signal and Communication Research Institute, China Academy of Railway Sciences Corporation Limited, Beijing 100081, China
[*2]The Center of National Railway Intelligent Transportation System Engineering and Technology, China Academy of Railway Sciences Corporation Limited, Beijing 100081, China
[*3]China Academy of Railway Sciences Corporation Limited, Beijing 100081, China
[*4]Postgraduate Department, China Academy of Railway Sciences, Beijing 100081, China
[*5]School of Automation, Beijing Institute of Technology, Beijing 100081, China
[*6]Key Laboratory of Intelligent Control and Decision of Complex Systems, Beijing Institute of Technology, Beijing 100081, China
E-mail: 13701193534@139.com, brucebin@bit.edu.cn
[†]Corresponding authors
[Received 00/00/00; accepted 00/00/00]

This article addresses a high-speed railway train timetable rescheduling (TTR) problem with a complete blockage at the station and train operation constraints. The problem is formulated as a mixed-integer linear programming (MILP) model minimizing the weighted sum of the total delay time of trains. A memetic algorithm (MA) is proposed, and the individual of MA is represented as the permutation of trains' departure order at the disrupted station. The individual is decoded to a feasible schedule of trains using a rule-based method to allocate the running time in sections and dwell time at stations. As a result, the original problem is reformulated as an unconstrained one. Several permutation-based operators are involved, including crossover, mutation, and local search. A restart strategy is employed to maintain the diversity of the population. The proposed MA is compared with first-scheduled-first-served (FSFS) and other state-of-the-art evolutionary algorithms. Experimental results demonstrate the superiority of MA in solving TTR through permutation-based optimization in terms of constraint handling, solution quality, and computation time.

Keywords: high-speed railway, train timetable rescheduling, disruptions, memetic algorithm, combinatorial optimization

1. Introduction

High-speed railway (HSR) plays an important role in medium-to-long distance transportation service in China. HSR is operating according to the prescribed timetable. However, HSR may face inevitable emergencies, e.g., infrastructure failure, train failure, and natural disasters [1]. Train operations may be disturbed/disrupted with delays. Therefore, train timetable rescheduling (TTR) is required for trains to recover to their regular operation.

A variety of studies have been analyzed for the TTR problem, which is proven to be NP-hard [2, 3]. In most studies, a mixed-integer linear programming (MILP) model is adopted, and the CPLEX solver is used to obtain solutions. A MILP model was proposed in [4] to deal with the real-time rescheduling of the timetable in case of a complete blockage in a railway segment. However, when the scale of the problem is getting larger, using the CPLEX solver will cost a lot of time, which may exceed the time limit.

Metaheuristics are usually used for solving NP-hard problems [5]. Near-optimal solutions are obtained with limited time. Genetic algorithm-based particle swarm optimization had been used to reschedule the HSR timetable under primary delays [6]. Meng *et al.* [7] considered train rescheduling with track assignment and proposed an artificial bee colony algorithm to solve the problem. Departure time and arrival time of trains are used for solution representation. However, this may obtain constraint violated solutions during the process. Some related works considered determining the order of trains to determine the train timetable. Wang *et al.* [8] adjusted train departure sequences based on Monte Carlo tree search. Wang and Wang [9] proposed an effective estimation of distribution algorithm (EDA) to solve the multi-track train scheduling problem. The permutation of train priority was obtained. Ding *et al.* [10] used several metaheuristics to solve the TTR problem as a permutation-based combinatorial optimization. However, the proposed metaheuristics are not well designed for the permutation-based optimization problem.

The memetic algorithm (MA) is a population-based metaheuristic that combines the evolutionary algorithms (EA) and local search techniques [11]. It has been adopted in solving many complex optimization problems [12–15]. The encoding scheme, genetic operator (i.e., the selection, crossover, and mutation operators), and local search strategy are important for the performance of MA. Some of them are specially designed for the permutation-based optimization problem.

The main contributions of this study are summarized as follows. First, the high-speed railway train timetable rescheduling problem with a complete station blockage is proposed and modeled as a MILP problem. Second, an effective permutation encoding method is proposed for the TTR problem, and a rule-based decoding method is designed to obtain a new schedule. These encoding and decoding methods can manage the entire constraints and guarantee the feasibility of the solution. Then, a MA is proposed to solve the permutation-based optimization problem with various permutation-based operators. A local search operator is developed to exploit the neighborhood of the best current individual, and a restart strategy is used to maintain the diversity of the population. Finally, experimental results show that the proposed MA can efficiently solve most test instances compared with state-of-the-art algorithms.

The rest of this paper is organized as follows. The TTR problem is described in Section 2. Section 3 presents a memetic algorithm to solve TTR. The performances of the proposed algorithms are evaluated in Section 4. Finally, conclusions and future work are provided in Section 5.

2. Problem Formulation

Punctuality is an important factor for railway operations. However, in the case of disruption, the railway system may fall into disorder. Trains may not be able to arrive or depart at stations.

In this section, we introduce a MILP model to formulate the TTR problem. We need to determine the new arrival and departure time of the trains at stations in order to recover the railway operations.

There are seven assumptions: (1) All trains should follow their original schedules before disruption happens. (2) No trains are canceled in the train timetable rescheduling problem. (3) A macroscopic model is presented without considering the signaling systems and the station capacity. (4) The disruption considered is a complete blockage at the first station. All affected trains should depart after the disruption ends. (5) There is only one disruption whose duration is a known value. (6) Train reordering is not allowed except for departure trains at the first station. (7) Running time and dwell time supplements are not considered in the original timetable.

2.1. Indices

i, j: the index of train, $i, j \in T$

s: the index of station, $s \in S$

$(s, s+1)$: the index of section, which is between stations s and $s+1$, $(s, s+1) \in K$

s^*: the index of the disrupted station, $s^* \in S$

$O(i), D(i)$: the index of origin station and destination station of train i, respectively

2.2. Parameters

T: the set of trains

S: the set of stations

K: the set of sections

$T^a_{i,s}$: the arrival time of train i at station s in the orignal schedule

$T^d_{i,s}$: the departure time of train i at station s in the orignal schedule

$d_{i,s}$: the minimum dwell time at station s for train i

$Y_{i,s}$: the train stop indicator in the orignal schedule, 1 if train i stops at station s; 0 otherwise

$r^{min}_{i,(s,s+1)}$: the minimum running time at section $(s, s+1)$ for train i

$r^s_{i,(s,s+1)}$: the additional time caused by starting for train i in section $(s, s+1)$

$r^e_{i,(s,s+1)}$: the additional time caused by stopping for train i in section $(s, s+1)$

$h_{(s,s+1)}$: the minimal headway between two consecutive trains of the same direction on section $(s, s+1)$

w_i: the weight value for train i

M: a large positive number

H^s_{dis}: the start time of the disruption

D_{dis}: the duration of the disruption

2.3. Decision Variables

$t^a_{i,s}$: the actual arrival time of train i at station s

$t^d_{i,s}$: the actual departure time of train i at station s

$q_{i,j,(s,s+1)}$: the actual traversing order, 1 if train i traverses on section $(s, s+1)$ before train j; 0 otherwise

$y_{i,s}$: the actual train stop indicator, 1 if train i stops at station s; 0 otherwise

2.4. Mathematical Formulation

The optimization model for the TTR problem is a MILP model, which can be formulated as follows:

$$\min\ F = \sum_{i \in T} \sum_{s \in S} w_i (t^a_{i,s} - T^a_{i,s} + t^d_{i,s} - T^d_{i,s}) \quad (1)$$

$$\text{s.t. } t^d_{i,s} - t^a_{i,s} \geq d_{i,s}\ \forall i \in T; s \in S \quad (2)$$

$$t^a_{i,s+1} - t^d_{i,s} \geq r^{min}_{i,(s,s+1)} + r^s_{i,(s,s+1)} y_{i,s} + r^e_{i,(s,s+1)} y_{i,s+1}$$
$$\forall i \in T; s \in S \backslash D(i) \quad (3)$$

$$t^d_{j,s} - t^d_{i,s} \geq h_{(s,s+1)} q_{i,j,(s,s+1)} - M(1 - q_{i,j,(s,s+1)})$$
$$\forall i, j \in T; i \neq j; s \in S \backslash D(i) \quad (4)$$

$$t^a_{j,s+1} - t^a_{i,s+1} \geq h_{(s,s+1)} q_{i,j,(s,s+1)} - M(1 - q_{i,j,(s,s+1)})$$
$$\forall i, j \in T; i \neq j; s \in S \backslash D(i) \quad (5)$$

$$q_{i,j,(s,s+1)} + q_{j,i,(s,s+1)} = 1\ \forall i, j \in I; i \neq j; s \in S \backslash D(i) \quad (6)$$

$$t^a_{i,s} = T^a_{i,s}\ \forall i \in T; s \in S : T^a_{i,s} \leq H^s_{dis} \quad (7)$$

$$t^d_{i,s} = T^d_{i,s}\ \forall i \in T; s \in S : T^d_{i,s} \leq H^s_{dis} \quad (8)$$

$$t^a_{i,s^*} \geq H^s_{dis} + D_{dis} \ \forall i \in T : H^s_{dis} \leq T^a_{i,s^*} \leq H^s_{dis} + D_{dis} \quad (9)$$

$$t^a_{i,O(i)} = t^d_{i,O(i)} \ \forall i \in T \quad (10)$$

$$t^a_{i,s} \geq T^a_{i,s} \ \forall i \in T; s \in S \quad (11)$$

$$t^d_{i,s} \geq T^d_{i,s} \ \forall i \in T; s \in S \quad (12)$$

$$q_{i,j,(O(i),O(i)+1)} = q_{i,j,(s,s+1)}$$
$$\forall i,j \in T; i \neq j; s \in S \backslash \{O(i), D(i)\} \quad (13)$$

$$y_{i,s} \leq t^d_{i,s} - t^a_{i,s} \ \forall i \in T; s \in S \backslash \{O(i), D(i)\} \quad (14)$$

$$y_{i,s} \geq \frac{t^d_{i,s} - t^a_{i,s}}{M} \ \forall i \in T; s \in S \backslash \{O(i), D(i)\} \quad (15)$$

$$y_{i,s} \geq Y_{i,s} \ \forall i \in T; s \in S \backslash \{O(i), D(i)\} \quad (16)$$

$$y_{i,s} = Y_{i,s} \ \forall i \in T; s \in \{O(i), D(i)\} \quad (17)$$

$$t^a_{i,s}, t^d_{i,s} \geq 0 \ \forall i \in T; s \in S \quad (18)$$

$$q_{i,j,(s,s+1)} \in \{0,1\} \ \forall i,j \in T; i \neq j; s \in S \backslash D(i) \quad (19)$$

$$y_{i,s} \in \{0,1\} \ \forall i,j \in T; i \neq j; s \in S \quad (20)$$

where Eq. (1) is to minimize the weighted sum of total delay time, including the delay arrival and departure time of each train at all the stations. Eq. (2) is the minimum dwelling time constraint. Eq. (3) is the minimum running time constraint. Eqs. (4) and (5) are the headway constraints for departure headway and arrival headway, respectively. Eq. (6) is the traverse order constraint of two trains in a section, which means that either train i traverses on section$(s, s+1)$ before train j or later than train j. Eqs. (7) and (8) guarantee the arrival and departure times for the unaffected trains are equal to the original timetable, respectively. Eq. (9) guarantees that no trains are allowed to arrive at stations during the disruption. Eq. (10) means the arrival time and departure time are the same for the origin station. Eqs. (11) and (12) are the timetable constraints that restrict trains are not allowed to arrive and depart from stations before the original arrival and departure time, respectively. Eq. (13) guarantees that the actual traversing orders of all trains are equal to the traversing orders in their first section. Eqs. (14) to (17) are the train stop indicator constraints. Eqs. (18) to (20) restrict the decision variables to be real numbers and binary numbers.

3. Memetic Algorithm for TTR

Since the TTR problem is NP-hard, there is no polynomial-time algorithm to obtain the exact solution. In this section, a MA is presented for solving TTR. First, encoding and decoding are introduced to transform the original MILP problem into a permutation-based combinatorial optimization problem without constraints. Then, the proposed MA will randomly generate the initial population as a set of permutations. The population will be updated by crossover, mutation, local search, and restart operators iteratively. MA adopts different search methodologies, mainly including population-based search and local search techniques [11]. The evolutionary process is similar to the genetic algorithm (GA) for the population-based search, while a local search is developed. When there is no improvement in the population, the population restart will be performed. The process of MA is shown in **Algorithm 1**.

Algorithm 1 The memetic algorithm for TTR

Input: The original timetable information; The disruption information; The set of affected trains T_{dis}
Output: The actual arrival time $t^a_{i,s}$ and departure time $t^d_{i,s}$; The total delay time F

```
Generate the initial population pop randomly.
Set NFE = |pop|.
while NFE < MaxFEs do
    Select parent individuals through roulette wheel selection.
    Update pop through modified order crossover.
    Update pop through swap mutation.
    Update NFE according to the number of individuals for mutation.
    Merge the new populations with the original ones and obtain the best individuals according to the population size.
    Update pop through local search using Algorithm 3.
    Update NFE according to the iterations of the local search.
    if the number of different individuals in the population pop is less than the predefined threshold σ then
        Regenerate the population pop randomly.
        NFE = NFE + |pop|.
    end if
end while
Find the best inidividual p through the evolution process in pop and decode it using Algorithm 2.
return t^a_{i,s}, t^d_{i,s}, and F.
```

3.1. Solution Representation

For TTR, most studies use the real-coded encoding scheme. The arrival and departure times are used as the solution. However, it is easy to obtain constraint violations during the evolutionary computation process. Suppose the traversing order of trains in each section is determined. In that case, we only need to figure out the arrival and departure times that satisfy the operation constraints, e.g., dwelling time, running time, headway constraints, etc. In this section, we propose a permutation-based encoding method for solving TTR. The problem for determining the traversing order of trains is an unconstrained one, which is easier than the original MILP problem. The integer number in the solution determines the rescheduling order of the trains. For example, a solution $\mathbf{p} = (1,2,4,3,5)$ represents the order of 5 trains, where train 4 is scheduled first before train 3. The order for the other trains remains the same. Since before disruption happens, trains follow their original schedules, the set of affected trains T_{dis} can be determined if the arrival time

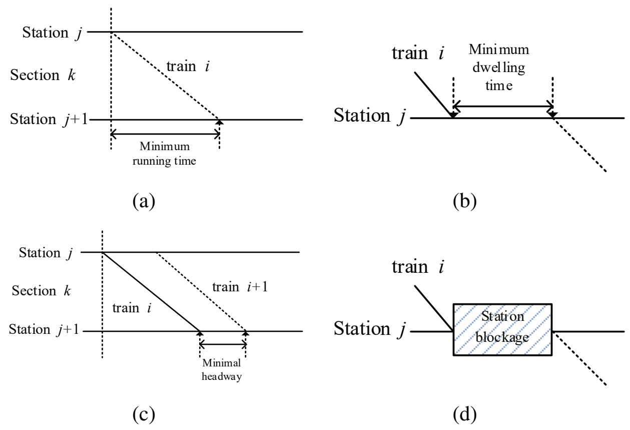


Fig. 1. Determine arrival and departure time in the decoding procedure. (a) Minimum running time constraints. (b) Minimum dwelling time constraints. (c) Headway constraints. (d) Depart after disruption ends.

at the first station is after H^s_{dis}. Therefore, $|T_{dis}|$ is the dimension of the permutation-based optimization problem. For the problem using real-coded encoding scheme, the dimension is $2 \cdot |T_{dis}| \cdot |S|$. The dimension of the problem using a permutation-based encoding scheme has been greatly decreased compared with that using a real-coded encoding scheme. Besides, the searching range for each element has also been decreased. It is decreased from 1440 (minutes of one day) to $|T_{dis}|$ (number of affected trains).

We could obtain the actual arrival time and departure time through the decoding procedure shown in **Algorithm 2** for a permutation encoded solution $\mathbf{p} = (p_1, p_2, \ldots, p_{|T|})$. Besides, the feasibility of the solution decoded from the permutation can be guaranteed since the constraint handling technique is implied during encoding and decoding. It follows a rule that trains may arrive and depart at stations once they are allowed as soon as possible. To better illustrate the constraints handling process, some of the constraints are described in **Fig. 1**. For example, the minimum running time constraints determine the arrival time. The minimum dwelling time constraints determine the departure time. Headway constraints determine both arrival and departure times. Trains should depart after the disruption ends.

Remark 1: In **Algorithm 2**, all constraints for TTR are satisfied. In line 22, the condition will be met when an addition stop may be added at station s. It may add to the total running time for section $(s-1, s)$ because of additional time caused by stopping. Therefore, the arrival time should be updated. If the arrival time is larger than the departure time, adding stop is canceled, and the arrival time is set to the departure time.

3.2. Selection Operator

The operator used for selection is the roulette wheel selection. It is usually used in GA. The individuals are selected according to their fitness values. Since it is a problem for minimization, the probabilities of the individuals are set according to the exponential of the negative fitness values.

Algorithm 2 Decoding Procedure

Input: The original timetable information; The disruption information; The set of affected trains T_{dis}; Scheduling order of the trains $\mathbf{p} = [p_i]_{1\times|T|}$

Output: The actual arrival time $t^a_{i,s}$ and departure time $t^d_{i,s}$

1: **for** $i = 1$ to $|T| - |T_{dis}|$ **do**
2:     **for** $s = O(i)$ to $D(i)$ **do**
3:         $t^a_{i,s} = T^a_{i,s}$; $t^d_{i,s} = T^d_{i,s}$;
4:     **end for**
5: **end for**
6: **for** $i = |T| - |T_{dis}| + 1$ to $|T|$ **do**
7:     **if** $i = |T| - |T_{dis}| + 1$ **then**
8:         $t^a_{p_i,O(p_i)} = H^s_{dis} + D_{dis}$;
9:         $t^d_{p_i,O(p_i)} = t^a_{p_i,O(p_i)}$;
10:     **else**
11:         $t^a_{p_i,O(p_i)} = \max(t^a_{p_{i-1},O(p_{i-1})} + h_{(O(p_i),O(p_i)+1)}, T^a_{p_i,O(p_i)})$;
12:         $t^d_{p_i,O(p_i)} = \max(t^a_{p_i,O(p_i)} + d_{p_i,O(p_i)}, T^d_{p_i,O(p_i)})$;
13:     **end if**
14:     $y_{p_i,O(p_i)} = Y_{p_i,O(p_i)}$;
15:     **for** $s = O(i) + 1$ to $D(i)$ **do**
16:         $y_{p_i,s} = Y_{p_i,s}$;
17:         $t^a_{p_i,s} = \max(t^d_{p_i,s-1} + r^{min}_{p_i,(s-1,s)} + y_{p_i,s-1} r^s_{p_i,(s-1,s)} + y_{p_i,s} r^e_{p_i,(s-1,s)}, T^a_{p_i,s})$;
18:         $t^a_{p_i,s} = \max(t^a_{p_i,s}, t^a_{p_{i-1},s} + h_{(s-1,s)})$;
19:         $t^d_{p_i,s} = \max(t^a_{p_i,s} + d_{p_i,s}, T^d_{p_i,s})$;
20:         **if** $s < D(p_i)$ **then**
21:             $t^d_{p_i,s} = \max(t^d_{p_i,s}, t^d_{p_{i-1},s} + h_{(s,s+1)})$;
22:             **if** $\mathrm{sgn}(t^d_{p_i,s} - t^a_{p_i,s}) > y_{p_i,s}$ **then**
23:                 $t^a_{p_i,s} = \min(t^d_{p_i,s-1} + r^{min}_{p_i,(s-1,s)} + y_{p_i,s-1} r^s_{p_i,(s-1,s)} + r^e_{p_i,(s-1,s)}, t^d_{p_i,s})$;
24:                 $y_{p_i,s} = \mathrm{sgn}(t^d_{p_i,s} - t^a_{p_i,s})$;
25:             **end if**
26:         **end if**
27:     **end for**
28: **end for**
29: **return** $t^a_{i,s}$ and $t^d_{i,s}$.

3.3. Crossover Operator

The modified order crossover (MOC) is adopted for the proposed MA [16]. It is designed for permutation-based combinatorial optimization problems, e.g., traveling salesman problem. The MOC operator randomly selects a crossover point to divide both parent individuals p_1 and p_2 into left and right strings with the same length. Then, the order of the right string p_1 is used to change the order of the positions in p_2 and vise versa. **Fig. 2** shows an example of MOC. The MOC operator is employed to produce offsprings of two parent individuals based on the

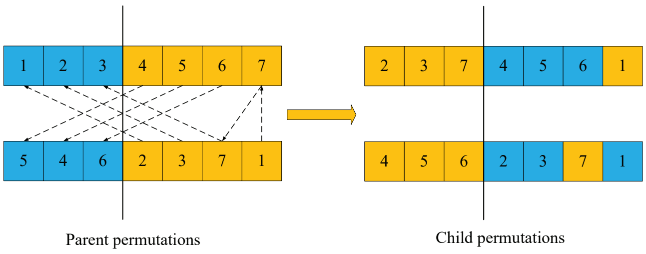


Fig. 2. Modified order crossover.

Algorithm 3 Local Search

Input: Current population *pop*; Best individual p_{best}; Number of iterations of local search L; Fitness function of total delay time F

Output: Updated population *pop*

for $i = 1$ to L **do**
    Perform swap operator on the best individual p_{best} to obtain a new individual $p(i)$.
    Calculate the objective value $F(i)$ for the new individual $p(i)$.
end for
Find the best individual $p(i_{best})$ based on $F(i)$.
Replace the worst one in the current population *pop* by $p(i_{best})$.
return *pop*

---

crossover rate p_c.

3.4. Mutation Operator

The mutation will be conducted after crossover. It should be mentioned that not all the newly obtained individuals are mutated. It is based on the mutation rate p_m. The mutation operator helps maintain the population's diversity by changing some of the individuals in the current population. The swap operator is chosen for mutation. We randomly select two positions in an individual and swap them to obtain a new permutation.

3.5. Survivor Selection

The survivor selection operator selects the fittest individuals who remain in the population. The truncation selection selects the top solutions with the population size based on the objective values.

3.6. Local Search

Local search is an important process in MA, which helps maintain the tradeoff between exploration and exploitation. The best individual in the population is used for local search. There are L iterations for local search. The best individual obtained in the local search process will replace the worst individual in the population. The local search process is described in **Algorithm 3**.

3.7. Restart Strategy

During the evolutionary process of MA, the population will converge to similar ones, which greatly decreases its diversity. As a result, it is difficult to generate new solutions. The restart strategy is employed to reduce wasted time and improve the diversity of the population. When the population updates, the objective values of the new individuals are calculated. If the number of different objective values in the new population is below a predefined threshold σ, the whole population will be reinitialized randomly.

4. Computational Experiments

This section presents the performance investigation of the proposed algorithms. At first, we present the test instances for TTR. Then, we solve the problem under different methods, including exact solutions by CPLEX. All experiments were carried out on a PC with an Intel Xeon Gold 5218 CPU 2.30GHz and 32 GB internal memory. Exact solutions for TTR problems were implemented in MATLAB R2019b using YALMIP as the modeling language and CPLEX 12.10 with default parameter settings [17]. Other algorithms for TTR problems were implemented in MATLAB R2019b.

4.1. Test Instances for TTR

Due to the lack of benchmark instances with disruptions for TTR in literature, we first develop the test instances. The Beijing–Tianjin intercity railway timetable from Beijing South to Tianjin is considered in this paper. There are altogether 6 stations and 5 sections. 40 trains downstream from 6:00 to 12:00 are considered for the railway timetable. The minimum dwell time for train stops at stations is set to 2 min and no dwell time for pass-through stations, the origin stations, and destination stations. The minimum running time of each section is shown in **Table 1**. The additional times caused by starting and stopping are set to 2 min and 3 min, respectively. The minimal headway is set to 4 min. The start time of the disruption H^s_{dis} is set to 6:40. s^* is set to 1, which means the disruption is at the first station. M is set to 1440 min.

We categorize the generation of w_i into the following two cases:

Case 1: The weight values w_i of trains are set to 1.

Case 2: The weight values w_i of trains are generated as uniformly distributed random integers in a range between 1 to 10.

To validate the performance of the algorithms, we produce 8 test instances. The first four instances (No. 1 – 4) are from Case 1, and the last four instances (No. 5 – 8) are from Case 2. The settings of the two basic parameters T and D_{dis} are listed in **Table 2**. For instances with the number of trains T less than 40, e.g., instance No. 1, the first train is the same train starting from 6:00. We do not need to adjust the schedule of all 40 trains when the duration of the disruption is only 30 min. As a result, T for

Table 1. The minimum running time in each section between two stations.

No.	Section	Time (min)
1	Beijing South - Yizhuang	5
2	Yizhuang - Yongle	5
3	Yongle - Wuqin	6
4	Wuqin - Nancang	5
5	Nancang - Tianjin	5

Table 2. Setting of the two basic parameters for the test instances.

No.	$\|T\|$	D_{dis} (min)	No.	$\|T\|$	D_{dis} (min)
1, 5	15	30	2, 6	20	50
3, 7	30	70	4, 8	40	90

different instances are generated according to the duration of the disruption D_{dis}.

4.2. Algorithms for Comparison

To evaluate the performance of the proposed MA, we compare the proposed MA with the following 6 algorithms, including First-scheduled-first-served (FSFS), dual-model estimation of distribution algorithm (DM-EDA) [18], Comprehensive learning particle swarm optimizer (CLPSO) [19], Covariance Matrix Adaptation Evolution Strategy (CMA-ES) [20], and genetic algorithm (GA) [21]. For GA, we apply the roulette wheel selection, modified order crossover, and swap mutation, which is similar to the proposed MA without local search and restart strategy.

Remark 2: Since SaDE, CLPSO, and CMA-ES are algorithms designed to search in continuous space, the random key algorithm is applied to transform the real-valued vector to a permutation. Given a real-valued vector (3.5, 2.4, 1.6, 0.5, 4.1), the permutation obtained is the ranking of the real-valued vector, which is (4, 3, 2, 1, 5). The range for each element in the real-valued vector is also the dimension of the vector.

4.3. Parameter Settings

For all the algorithms, the particle/population size is set to $10 \cdot D$, where D is the dimension of the searching space. For DM-EDA, the subpopulation size N_{adv} is set to D, which is 10% of the population size. The learning rate μ_n and μ_e are both set to 0.2. The predefined threshold ε is set to 0.01. For CLPSO, the acceleration constant c is set to 1.49445. The inertia weight w is selected linear decreasing from 0.9 to 0.4. For GA and MA, the crossover rates p_c are both set to 0.9. The mutation rates p_m are both set to 0.05. For MA, the number of local search iterations L is set to 100. The predefined threshold σ is set to 2. Each algorithm is terminated when the maximum number of fitness evaluations ($10000 \cdot D$) is reached (i.e., $MaxFEs = 10000 \cdot D$). The independent runs for each algorithm on each instance are set to 20.

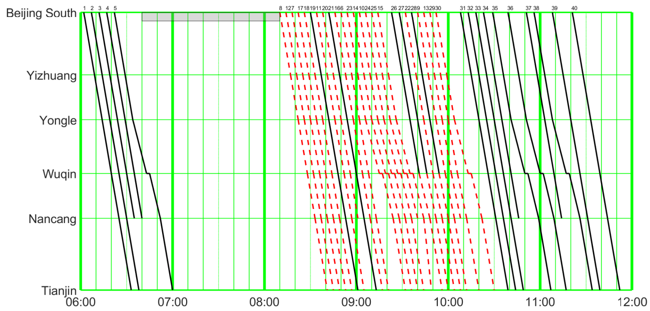


Fig. 3. Rescheduled train timetable for instance No. 8.

Most of the algorithms' parameters are kept the same as the original papers. Besides, for the CPLEX solver, the termination time is set to 3600s.

4.4. Results and Analysis

For the proposed MA, the best settings for the number of local search iterations L and the predefined threshold σ for restart strategy are not determined. To analyze the sensitivity of L and σ, we test MA in all test instances. **Table 3** shows the results of MA with different parameters (L, σ). In the table, the mean value and standard deviation of MA in 20 independent runs are shown. In seven instances (No. 1 – 7), MA performs well with different parameters. For most parameters, the standard deviations are zero, which means the parameters are not sensitive. For instance No. 8, the best result is indicated in **bold**. It shows that the number of local search iterations L is set to 100, and the predefined threshold σ for restart strategy is set to 2.

Based on the given parameter settings, we compare the proposed MA with 6 algorithms and CPLEX. **Table 4** shows the result of 20 independent runs of each algorithm with mean values and standard deviations. For CPLEX, it only runs once. The best results are indicated in **bold**.

It can be drawn from **Table 4** that the proposed MA outperforms other methods. In seven instances (No. 1 – 7), the results of MA equal that of CPLEX. Among these instances, the results of MA are proven to be optimal in instances No. 1, 2, 5, 6, and 7. Moreover, for instances No. 3 and 4, the results of MA are equal to CPLEX (stopped within one hour). In instance No. 8, the result of MA is better than that of CPLEX (stopped within one hour). **Fig. 3** shows the rescheduled train timetable for instance No. 8 obtained by MA with the objective value 43122. From the figure, disrupted trains (dotted lines) with fewer stops are scheduled earlier than those trains that stop at Wuqin.

In instances No. 1 and 2, all algorithms except for FSFS converge to the optimal value. It is because the size of the instance is small, and the algorithms can cover nearly all feasible solutions. As for FSFS, the order of the trains is kept the same, which means the original order is not optimal under disruption. SaDE also shows good performance in several instances, but it is inferior to MA. It is because SaDE is not designed for permutation-based optimization. A permutation-based MA with local search

Table 3. Results of MA with different parameters (L, σ).

No.	(20, 2)	(20, 3)	(20, 4)	(50, 2)	(50, 3)	(50, 4)	(80, 2)	(80, 3)	(80, 4)	(100, 2)	(100, 3)	(100, 4)
1	1628.0000	1628.0000	1628.0000	1628.0000	1628.0000	1628.0000	1628.0000	1628.0000	1628.0000	1628.0000	1628.0000	1628.0000
	(0.0000)	(0.0000)	(0.0000)	(0.0000)	(0.0000)	(0.0000)	(0.0000)	(0.0000)	(0.0000)	(0.0000)	(0.0000)	(0.0000)
2	3874.0000	3874.0000	3874.0000	3874.0000	3874.0000	3874.0000	3874.0000	3874.0000	3874.0000	3874.0000	3874.0000	3874.0000
	(0.0000)	(0.0000)	(0.0000)	(0.0000)	(0.0000)	(0.0000)	(0.0000)	(0.0000)	(0.0000)	(0.0000)	(0.0000)	(0.0000)
3	7268.8000	7268.8000	7271.2000	7268.0000	7268.0000	7268.0000	7268.0000	7268.0000	7268.0000	7268.0000	7268.0000	7268.0000
	(2.4623)	(3.5777)	(5.4445)	(0.0000)	(0.0000)	(0.0000)	(0.0000)	(0.0000)	(0.0000)	(0.0000)	(0.0000)	(0.0000)
4	12070.3000	12070.3000	12070.6000	12070.0000	12070.0000	12070.0000	12070.0000	12070.0000	12070.0000	12070.0000	12070.0000	12070.0000
	(1.3416)	(1.3416)	(1.8468)	(0.0000)	(0.0000)	(0.0000)	(0.0000)	(0.0000)	(0.0000)	(0.0000)	(0.0000)	(0.0000)
5	6126.0000	6126.0000	6126.0000	6126.0000	6126.0000	6126.0000	6126.0000	6126.0000	6126.0000	6126.0000	6126.0000	6126.0000
	(0.0000)	(0.0000)	(0.0000)	(0.0000)	(0.0000)	(0.0000)	(0.0000)	(0.0000)	(0.0000)	(0.0000)	(0.0000)	(0.0000)
6	14810.0000	14810.0000	14810.0000	14810.0000	14810.0000	14810.0000	14810.0000	14810.0000	14810.0000	14810.0000	14810.0000	14810.0000
	(0.0000)	(0.0000)	(0.0000)	(0.0000)	(0.0000)	(0.0000)	(0.0000)	(0.0000)	(0.0000)	(0.0000)	(0.0000)	(0.0000)
7	26889.0000	26890.1000	26923.5500	26872.0000	26872.0000	26872.0000	26872.0000	26872.0000	26872.0000	26872.0000	26872.0000	26872.0000
	(45.3965)	(47.1971)	(68.9137)	(0.0000)	(0.0000)	(0.0000)	(0.0000)	(0.0000)	(0.0000)	(0.0000)	(0.0000)	(0.0000)
8	43156.4000	43161.0000	43228.6000	43124.1000	43125.0000	43129.5000	43123.2000	43136.3000	43123.5000	**43122.6000**	43128.9000	43122.9000
	(47.9104)	(58.1088)	(93.8714)	(2.9362)	(10.7508)	(15.3125)	(2.4623)	(56.9516)	(2.6656)	**(1.8468)**	(28.0936)	(2.1981)

Table 4. Results of the comparison on the objective value of different algorithms.

No.	FSFS	DM-EDA	SaDE	CLPSO	CMA-ES	GA	MA	CPLEX
1	1700.00	**1628.00 ± 0.00**‡	**1628.00 ± 0.00**‡	**1628.00 ± 0.00**‡	**1628.00 ± 0.00**‡	**1628.00 ± 0.00**‡	**1628.00 ± 0.00**‡	1628.00‡
2	3962.00	**3874.00 ± 0.00**‡	**3874.00 ± 0.00**‡	**3874.00 ± 0.00**‡	**3874.00 ± 0.00**‡	**3874.00 ± 0.00**‡	**3874.00 ± 0.00**‡	3874.00‡
3	7616.00	7570.80 ± 34.55	7272.80 ± 7.52	7274.40 ± 7.61	7284.30 ± 0.73	7277.50 ± 8.85	**7268.00 ± 0.00**	7268.00†
4	12554.00	12539.20 ± 55.02	**12070.00 ± 0.00**	12072.10 ± 3.34	12081.70 ± 13.27	12072.60 ± 8.00	**12070.00 ± 0.00**	12070.00†
5	8012.00	6462.00 ± 0.00	**6126.00 ± 0.00**‡	**6126.00 ± 0.00**‡	**6126.00 ± 0.00**‡	**6126.00 ± 0.00**‡	**6126.00 ± 0.00**‡	6126.00‡
6	17606.00	15386.00 ± 0.00	**14810.00 ± 0.00**‡	**14810.00 ± 0.00**‡	14852.80 ± 87.82	14852.80 ± 87.82	**14810.00 ± 0.00**‡	14810.00‡
7	35452.00	31475.05 ± 684.50	26874.60 ± 8.00	26875.30 ± 8.32	27177.00 ± 330.65	27038.70 ± 64.42	**26872.00 ± 0.00**‡	26872.00‡
8	61640.00	59492.10 ± 1055.76	43125.00 ± 10.75	43636.00 ± 157.02	43697.00 ± 599.01	43333.50 ± 253.86	**43122.30 ± 1.34**	43128.00†

† CPLEX stopped after running for one hour.

‡ Optimal value.

Table 5. Runtime performance of different algorithms (sec.).

No.	FSFS	DM-EDA	SaDE	CLPSO	CMA-ES	GA	MA	CPLEX
1	0.01	5.74 ± 0.46	8.18 ± 0.65	3.75 ± 0.30	2.24 ± 0.36	4.33 ± 0.20	4.18 ± 0.15	10.39
2	<0.01	10.09 ± 0.69	11.99 ± 0.62	5.57 ± 0.39	3.05 ± 0.27	6.83 ± 0.17	5.89 ± 0.15	64.75
3	<0.01	24.89 ± 0.97	19.91 ± 1.24	11.27 ± 1.80	6.07 ± 1.00	12.98 ± 0.28	11.77 ± 0.24	–
4	<0.01	47.55 ± 1.82	30.01 ± 2.13	17.36 ± 0.65	9.67 ± 0.19	20.46 ± 0.15	19.18 ± 0.38	–
5	0.01	5.12 ± 0.55	8.11 ± 1.17	3.71 ± 0.69	1.88 ± 0.31	4.33 ± 0.13	3.61 ± 0.14	10.55
6	<0.01	10.28 ± 1.29	12.31 ± 2.03	6.06 ± 0.71	2.76 ± 0.12	6.84 ± 0.20	6.05 ± 0.22	30.59
7	<0.01	24.89 ± 0.82	20.10 ± 1.15	11.31 ± 1.33	6.27 ± 1.10	12.87 ± 0.19	11.81 ± 0.20	2861.86
8	<0.01	49.87 ± 3.10	31.19 ± 2.93	17.41 ± 0.82	10.46 ± 1.63	20.55 ± 0.31	19.23 ± 0.31	–

– CPLEX cannot find optimal value after running for one hour.

mechanism and restart strategy shows its great effectiveness.

Figs. 4 and **5** provide the converge curves of different algorithms in instances No. 7 and 8. The curves are zoomed-in in some areas for better visualization. The horizontal and vertical axes represent the number of fitness evaluations and the mean of the objective function of 20 runs, respectively. It can be drawn from the figure that MA converges faster than other algorithms at the beginning. Besides, both GA and CMA-ES have a high convergence speed. In the end, the final result of MA is better than other algorithms.

Table 5 shows the running time of FSFS, EAs, and CPLEX. It shows the mean values and standard deviations of 20 independent running for EAs. The best results are indicated in **bold**. The result shows that SaDE takes more computation time in small-scale instances, while DM-EDA takes more computation time in large-scale instances among the EAs. For FSFS, all instances are solved within 0.01s. It can be seen that all instances can be solved within one minute. However, the running time for CPLEX increases a lot with the increase of the problem size. For instance No. 7, the total running time is around 2862s, and for instances (No. 3, 4, and 8), the total running time is more than 3600s. This result implies the efficiency of the proposed framework with permutation-based encoding and the rule-based decoding methods.

Based on the above performance results, we can see that the proposed MA has successfully solved most of the test instances for TTR and shows significant advantages compared with other algorithms. The main reasons are as follows:

(a) The permutation-based encoding scheme and rule-based decoding method significantly reduce the complexity of the problem. The encoding scheme significantly decreases the searching space, and the

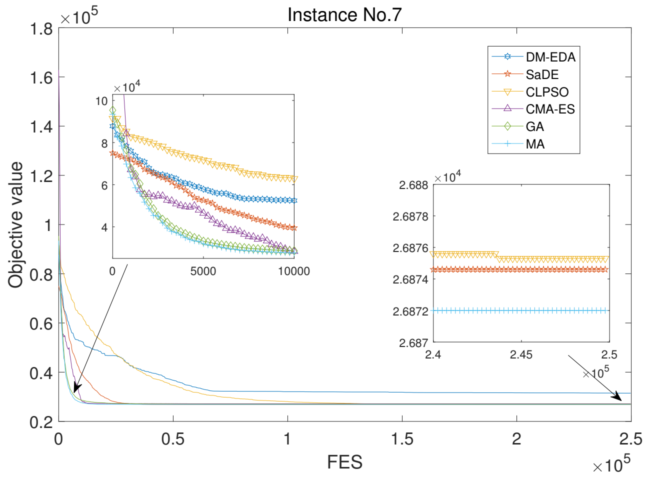


Fig. 4. Convergence curves of different algorithms for instance No. 7.

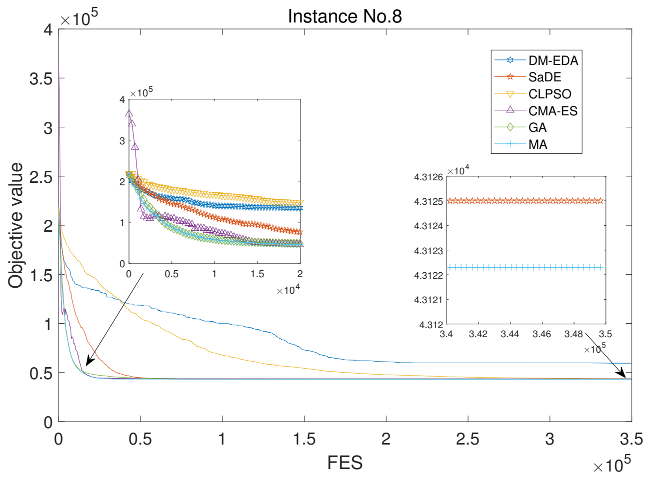


Fig. 5. Convergence curves of different algorithms for instance No. 8.

decoding method can guarantee the feasibility of the solution. Therefore, TTR can be solved within a limited computation time.

(b) The proposed MA is designed in a permutation-based encoding scheme. Selection, crossover, and mutation operators are designed for permutation-based combinatorial optimization problems. These make it more effective for TTR than other algorithms, especially those algorithms for continuous space with the random key algorithm to obtain the permutation.

(c) The local search improves the exploration ability of the proposed MA, and the restart strategy improves the diversity of the solution.

5. Conclusion

The high-speed railway TTR problem is formulated as a MILP problem. A MA is proposed to deal with it. A novel encoding and decoding method are specially designed for TTR, transferring the original problem to an unconstrained one. This avoids a large amount of ineffective search in the solution space. Except for crossover and mutation operators, local search strategy and restart strategy are applied to improve the searching ability. After being tested in 8 test instances, the proposed MA outperforms other algorithms and shows its efficiency compared with CPLEX. The results can be obtained within one minute, which is suitable for real-time rescheduling. In the future, we will consider situations with more types of trains (e.g., trains with different prefixes including G, C, D) and consider reordering at other stations based on the feature of the timetable. Besides, considering the uncertainties in the dynamic environment will make the model more practical [22]. The proposed MA can also be improved by using a constructive heuristic to obtain some good solutions for the initial population [11]. Meanwhile, the multi-objective TTR problem with more optimization objectives using metaheuristics deserves further research [23, 24].

Acknowledgements

This work was supported the National Natural Science Foundation of China under Grants U1834211, U1934220, 61822304, 62088101, and in part by the Foundation of China Academy of Railway Sciences Corporation Limited (No. 2021YJ043).

References:

[1] M. Zhou, H. Dong, X. Liu, H. Zhang, and F.-Y. Wang, "Integrated timetable rescheduling for multidispatching sections of high-speed railways during large-scale disruptions", IEEE Trans. Comput. Soc. Syst., 2021, in press.

[2] V. Cacchiani, D. Huisman, M. Kidd, L. Kroon, P. Toth, L. Veelenturf, and J. Wagenaar, "An overview of recovery models and algorithms for real-time railway rescheduling", Transp. Res. Part B Methodol., 63:15–37, 2014.

[3] W. Fang, S. Yang, and X. Yao, "A survey on problem models and solution approaches to rescheduling in railway networks", IEEE Trans. Intell. Transp. Syst., 16(6):2997–3016, 2015.

[4] S. Zhan, L. G. Kroon, L. P. Veelenturf, and J. C. Wagenaar, "Real-time high-speed train rescheduling in case of a complete blockage", Transp. Res. Part B Methodol., 78:182–201, 2015.

[5] S. Ding, C. Chen, Q. Zhang, B. Xin, and P. M. Pardalos, "Metaheuristics for resource deployment under uncertainty in complex systems", CRC Press, Boca Raton FL, USA, 1 edition, 2021.

[6] M. Wang, L. Wang, X. Xu, Y. Qin, and L. Qin, "Genetic algorithm-based particle swarm optimization approach to reschedule high-speed railway timetables: A case study in China", J. Adv. Transp., 2019, 2019.

[7] X. Meng, Y. Wang, W. Xiang, and L. Jia, "An integrated model for train rescheduling and station track assignment", IET Intell. Transp. Syst., 15(1):17–30, 2021.

[8] R. Wang, M. Zhou, Y. Li, Q. Zhang, and H. Dong, "A timetable rescheduling approach for railway based on monte carlo tree search", In 2019 IEEE Intell. Transp. Syst. Conf., pp. 3738–3743. IEEE, 2019.

[9] S. Wang and L. Wang, "An effective estimation of distribution algorithm for multi-track train scheduling problem", In Int. Conf. Intell. Comput., pp. 697–708. Springer, 2014.

[10] S. Ding, T. Zhang, R. Wang, C. Zhang, S. Lu, and B. Xin, "A comparative study on evolutionary algorithms for high-speed railway train timetable rescheduling problem", In 7th Int. Work. Adv. Comput. Intell. Intell. Informatics, 2021.

[11] F. Neri and C. Cotta, "Memetic algorithms and memetic computing optimization: A literature review", Swarm Evol. Comput., 2:1–14, 2012.
[12] Y. Wang, B. Xin, L. Dou, and Z. Peng, "A heuristic initialized memetic algorithm for the joint allocation of heterogeneous stochastic resources", In 2019 IEEE Congr. Evol. Comput., pp. 1929–1936. IEEE, 2019.
[13] Y. Ding, B. Xin, H. Zhang, and J. Chen, "A memetic algorithm for curvature-constrained path planning of messenger UAV in air-ground coordination", In 2020 IEEE Int. Conf. Syst. Man, Cybern., pp. 1465–1472. IEEE, 2020.
[14] G. Gao, Y. Mei, B. Xin, Y.-H. Jia, and W. Browne, "A memetic algorithm for the task allocation problem on multi-robot multi-point dynamic aggregation missions", In 2020 IEEE Congr. Evol. Comput., pp. 1–8. IEEE, 2020.
[15] Y. Wang, B. Xin, and J. Chen, "An adaptive memetic algorithm for the joint allocation of heterogeneous stochastic resources", IEEE Trans. Cybern., 2021, in press.
[16] A. J. Umbarkar and P. D. Sheth, "Crossover operators in genetic algorithms: a review", ICTACT J. Soft Comput., 6(1):1083–1092, 2015.
[17] J. Löfberg, "YALMIP: A toolbox for modeling and optimization in MATLAB", In Proc. CACSD Conf., Taipei, Taiwan, 2004.
[18] S. Lu and B. Xin, "Tabu-model-based estimation of distribution algorithm framework for permutation optimization problems", In 9th Int. Symp. Comput. Intell. Ind. Appl., 2020.
[19] J. J. Liang, A. K. Qin, P. N. Suganthan, and S. Baskar, "Comprehensive learning particle swarm optimizer for global optimization of multimodal functions", IEEE Trans. Evol. Comput., 10(3):281–295, 2006.
[20] N. Hansen and A. Ostermeier, "Adapting arbitrary normal mutation distributions in evolution strategies: The covariance matrix adaptation", In Proc. IEEE Int. Conf. Evol. Comput., pp. 312–317. IEEE, 1996.
[21] S. Mirjalili, "Genetic algorithm", In Evolutionary algorithms and neural networks, pp. 43–55, Springer, 2019.
[22] S. Ding, R. Wang, X. Zhou, Y. Ren, and K. Huang, "High-speed railway train timetable rescheduling in case of a stochastic section blockage", In 2021 Chinese Autom. Congr. IEEE, 2021.
[23] S. Ding, C. Chen, B. Xin, and P. M. Pardalos, "A bi-objective load balancing model in a distributed simulation system using NSGA-II and MOPSO approaches", Appl. Soft Comput., 63:249–267, 2018.
[24] W. Zheng, Y. Tan, M. Gao, W. Jia, and Q. Wang, "A modification of MOEA/D for solving multi-objective optimization problems", J. Adv. Comput. Intell. Intell. Inform., 22(2):214–223, 2018.

Name:
Shuxin Ding

Affiliation:
Associate Researcher, Signal and Communication Research Institute, China Academy of Railway Sciences Corporation Limited

Address:
No.2 Daliushu Road, Haidian District, Beijing 100081, China

Brief Biographical History:
2019- Assistant Researcher, Signal and Communication Research Institute, China Academy of Railway Sciences Corporation Limited
2021- Associate Researcher, Signal and Communication Research Institute, China Academy of Railway Sciences Corporation Limited

Main Works:
• S. Ding, C. Chen, B. Xin, P. M. Pardalos. "A bi-objective load balancing model in a distributed simulation system using NSGA-II and MOPSO approaches," Applied Soft Computing, Vol. 63, pp. 249–267, 2018.
• S. Ding, C. Chen, Q. Zhang, B. Xin, P. M. Pardalos. *Metaheuristics for Resource Deployment under Uncertainty in Complex Systems*, Boca Raton FL, USA: CRC Press, 2021.

Membership in Academic Societies:
• Chinese Association of Automation
• Institute of Electrical and Electronics Engineers (IEEE)

Name:
Tao Zhang

Affiliation:
Professor, Signal and Communication Research Institute, China Academy of Railway Sciences Corporation Limited

Address:
No.2 Daliushu Road, Haidian District, Beijing 100081, China

Brief Biographical History:
2020- Professor, Signal and Communication Research Institute, China Academy of Railway Sciences Corporation Limited

Main Works:
• T. Zhang, F. Chen, T. Wang, "High speed train rescheduling based on a cooperative particle swarm optimization algorithm," Journal of Information and Computational Science, Vol.11, No.17, pp. 6337–6343, 2014.

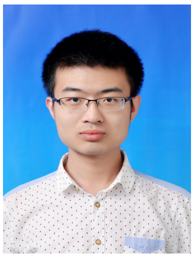
Name:
Ziyuan Liu

Affiliation:
Associate Researcher, China Academy of Railway Sciences Corporation Limited

Address:
No.2 Daliushu Road, Haidian District, Beijing 100081, China

Brief Biographical History:
2008-2012 Research Assistant, Signal and communication Research Institute, China Academy of Railway Sciences
2017- Associate Researcher, China Academy of Railway Sciences Corporation Limited

Main Works:
• Z. Liu, "Optimization of computer-aided decision-making system for railroad traffic dispatching command," Computer-Aided Design & Applications, Vol.19, No.S4, pp. 123–134, 2022.



Name:
Rongsheng Wang

Affiliation:
Ph.D. Student, Postgraduate Department, China Academy of Railway Sciences

Address:
No.2 Daliushu Road, Haidian District, Beijing 100081, China

Brief Biographical History:
2013-2017 B.S. Student, School of Information Science and Technology, Southwest Jiaotong University
2017-2020 M.S. Student, School of Electronic and Information Engineering, Beijing Jiaotong University
2020- Ph.D. Student, Postgraduate Department, China Academy of Railway Sciences

Main Works:
• R. Wang, M. Zhou, Y. Li, Q. Zhang, and H. Dong, "A timetable rescheduling approach for railway based on monte carlo tree search," 2019 IEEE Intelligent Transportation Systems Conference (ITSC), pp. 3738–3743. IEEE, 2019.

Name:
Sai Lu

Affiliation:
Ph.D. Student, School of Automation, Beijing Institute of Technology

Address:
No.5 Zhongguancun South Street, Haidian District, Beijing 100081, China

Brief Biographical History:
2013-2017 B.S. Student, School of Automation, Harbin Engineering University
2017-2019 M.S. Student, School of Automation, Beijing Institute of Technology
2019- Ph.D. Student, School of Automation, Beijing Institute of Technology

Main Works:
- S. Lu, B. Xin, L. Dou, and L. Wang, "A comparative study on evolutionary algorithms for the agent routing problem in multi-point dynamic task," International Journal of Automation and Control, Vol.14, No.5-6, pp. 571–592, 2020.
- S. Lu, B. Xin, H. Zhang, and J. Chen, "Agent-based self-organized constructive heuristics for travelling salesman problem," 2020 59th IEEE Conference on Decision and Control (CDC), pp. 1164–1169. IEEE, 2020.

Name:
Bin Xin

Affiliation:
Professor, State Key Laboratory of Intelligent Control and Decision of Complex Systems, Beijing Institute of Technology

Address:
No.5 Zhongguancun South Street, Haidian District, Beijing 100081, China

Brief Biographical History:
2012- Lecturer, School of Automation, Beijing Institute of Technology
2015- Associate Professor, School of Automation, Beijing Institute of Technology
2018- Professor, School of Automation, Beijing Institute of Technology

Main Works:
- B. Xin, J. Chen, J. Zhang, H. Fang, and Z.-H. Peng, "Hybridizing differential evolution and particle swarm optimization to design powerful optimizers: a review and taxonomy," IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews, Vol.42, No.5, pp. 744–767, 2012.
- Y. L. Ding, B. Xin, and J. Chen, "Curvature-constrained path elongation with expected length for Dubins vehicle," Automatica, Vol.108, Article No.108495, 2019.

Membership in Academic Societies:
- The Institute of Electrical and Electronics Engineers (IEEE) Computational Intelligence Society
- IEEE Control Systems Society
- Chinese Association of Automation
- Editorial Board of the Journal of Advanced Computational Intelligence and Intelligent Informatics (JACIII)

Name:
Zhiming Yuan

Affiliation:
Professor, Signal and Communication Research Institute, China Academy of Railway Sciences Corporation Limited

Address:
No.2 Daliushu Road, Haidian District, Beijing 100081, China

Brief Biographical History:
2019- Professor, Signal and Communication Research Institute, China Academy of Railway Sciences Corporation Limited

Main Works:
- Z. Yuan, L. Yan, Y. Gao, T. Zhang, S. Gao, "Virtual parameter learning-based adaptive control for protective automatic train operation," IEEE Transactions on Intelligent Transportation Systems, Vol.22, No.12, pp. 7943–7954, 2021.